

Report

Student_id: z5192086 Name: Pan Luo

DQN:

According to what I learned in lecture, I use Deep Q learning to implement this assignment. First, I define the network with liner regression, by adding two hidden layers and using Relu as activation function. Then I choose mean square as loss function because it always used in linear regression.

After defining the q_value and q_action , I can calculate $target_Q$ using Approximation Function. This function aims to approximate target q and can avoid state 'explode'. In this step, the model can run, but the reward is very low. Then I implement batching and experience replay. These two steps aim to save the previous $Q(state, action)$ which the model get. By saving these Q , the model can randomly choose different ways to go, and there is a hyper parameter called epsilon which means how much probability the model will choose a new way to learn. I implement a relay buffer to save the previous Q , if the size of replay buffer is bigger than replay size, then pop out 0 value in the replay buffer. If the size of replay buffer is bigger than batch size, then I can start to train. To make training stable, I adjust the hyper parameter many times.

Hyper parameter:

$GAMMA = 0.99$ $INITIAL_EPSILON = 0.5$ $FINAL_EPSILON = 0.01$

$EPSILON_DECAY_STEPS = 100$ $BATCH_SIZE = 128$ $REPLAY_SIZE = 30000$

$HIDDEN_NODES = 30$